编程练习8：

异常检测与推荐系统

机器学习

# 介绍

在本练习中，您将实现异常检测算法，并将其应用于检测网络上出现故障的服务器。在第二部分中，您将使用协作过滤来构建电影推荐系统。在开始编程练习之前，我们强烈建议观看视频讲座并完成相关主题的复习问题。

要开始练习，您需要下载起始代码并将其内容解压缩到要完成练习的目录。如果需要，请在开始本练习之前使用octave/matlab中的cd命令更改到此目录。

您也可以在课程网站的“环境设置说明”中找到安装octave/matlab的说明。

## 本练习中包含的文件

ex8.m-octave/Matlab脚本用于练习ex8的第一部分

ex8 cofi.m -octave/Matlab脚本用于练习

ex8data1.mat-异常检测的第一示例数据集

ex8data2.mat-异常检测的第二示例数据集

ex8 movies.mat-影评数据集

ex8 movieparams.mat-为调试多变量提供的参数

multivariateGaussian.m-计算高斯分布可视化拟合的概率密度函数。

visualizeFit.m-高斯分布的二维图和数据集。

checkCostFunction.m-协同过滤的梯度检查。

computeNumericalGradient.m-数值计算梯度

fmincg.m-函数最小化例程（类似于fminnc）

loadmovielist.m-将电影列表加载到单元格数组

movie ids.txt-电影列表。

normalizeRatings.m-协作筛选的平均规范化。

submit.m-将解决方案发送到服务器的提交脚本

[]estimateGaussian.m-使用对角协方差矩阵估计高斯分布的参数*？*

[]selectThreshold.m-查找异常检测阈值

[]coficostfunc.m-实现协同过滤的成本函数*？？*

*？*指示需要完成的文件

在练习的第一部分（异常检测）中，您将使用脚本ex8.m。在协作筛选的第二部分中，您将使用ex8 cofi.m。这些脚本为问题设置数据集，并调用将要编写的函数。您只需要按照此分配中的说明修改其他文件中的函数。

## 哪里能得到帮助

本课程中的练习使用Octaveor Matlab，一种非常适合数值计算的高级编程语言。如果您没有安装Octave或Matlab，请参阅课程网站“环境设置说明”中的安装说明。[〔1〕](" \l "_ftn1" \o ")

在octave/matlab命令行中，键入help后跟函数名，显示内置函数的文档。例如，“帮助绘图”将显示用于绘图的帮助信息。有关倍频程函数的更多文档可以在倍频程文档页面中找到。Matlab文档可以在Matlab文档页面找到。

我们也强烈鼓励利用网上讨论与其他学生讨论练习。但是，不要查看其他人编写的任何源代码，也不要与其他人共享您的源代码。

# 1             异常检测

在本练习中，您将实现异常检测算法来检测服务器计算机中的异常行为。这些特性测量每个服务器的吞吐量（MB/s）和响应延迟（ms）。当您的服务器运行时，您收集了m=307个它们的行为示例，因此有一个未标记的数据集{x（1），…，x（m）}。您怀疑这些示例中的绝大多数是服务器正常运行的“正常”（非异常）示例，但也可能有一些服务器在此数据集中异常运行的示例。

您将使用高斯模型来检测数据集中的异常示例。您将首先从二维数据集开始，该数据集将允许您可视化算法正在执行的操作。在该数据集上，您将拟合高斯分布，然后找到概率非常低的值，因此可以认为是异常。然后，将异常检测算法应用于具有多个维度的较大数据集。这部分练习你将使用ex8.m。

ex8.m的第一部分将可视化数据集，如图1所示。

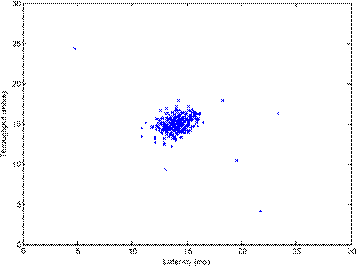


图1：第一个数据集。

## 1.1           高斯分布

要执行异常检测，首先需要根据数据的分布拟合一个模型。

给定一个训练集{x（1），…，x（m）}（其中x（i）∈r*n*，你要估计每个特征XI的高斯分布。对于每个特征i=1…n，您需要找到与每个示例的第i维-第h维中的数据相匹配的参数μi和σi2）。

高斯分布由

*,*

其中，μ是平均值，σ2控制方差。

## 1.2           高斯函数的参数估计

您可以使用以下方程估计第i个特征的参数（μi，σi2）。为了估计平均值，您将使用：

*,* （1）

对于差异，您将使用：

*.* （2）

你的任务是在EstimateGaussian.m中完成代码。此函数以数据矩阵X作为输入，并应输出一个N维向量mu（包含所有N个特征的平均值）和另一个N维向量sigma2（包含所有特征的方差）。您可以在每个特性和每个训练示例上使用for循环来实现这一点（尽管矢量化实现可能更有效；如果您愿意，可以随意使用矢量化实现）。注意，在octave/matlab中，var函数将（默认情况下）使用，而不是计算σi2。

一旦您完成了EstimateGaussian.m中的代码，ex8.m的下一部分将可视化拟合高斯分布的轮廓。您应该得到一个类似于图2的图。从图中可以看到，大多数示例位于概率最高的区域，而异常示例位于概率较低的区域。

*你现在应该提交你的解决方案。*

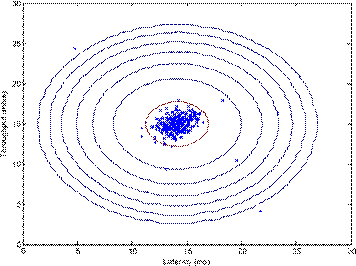


图2：分布的高斯分布轮廓符合数据集。

## 1.3           选择阈值，*ε*

现在您已经估计了高斯参数，您可以研究给定此分布的哪些示例具有非常高的概率，哪些示例具有非常低的概率。低概率的例子更可能是我们数据集中的异常。确定哪些示例是异常的一种方法是基于交叉验证集选择阈值。在本部分练习中，您将实现一个算法，使用交叉验证集上的f1分数选择阈值ε。

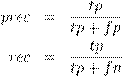
现在您应该在selectThreshold.m中完成代码. 为此，我们将使用交叉验证集，其中标签y=1对应于异常示例，y=0对应于正常示例。对于每个交叉验证示例，我们将计算）。所有这些概率的向量）传递给在向量pval中选择threshold.m。相应的标签ycv（1），…，ycv（错误地传递给向量yval中的同一个函数）。简历）

函数selectThreshold.m应返回两个值；第一个值是选定的阈值ε。如果示例x具有低概率p（x）<ε，则将其视为异常。这个函数还应该返回f1分数，它告诉你在给定某个阈值的情况下，你在寻找地面真值异常方面做得有多好。对于许多不同的ε值，您将通过计算当前阈值正确和错误分类的示例数来计算得到的f1分数。

F1分数使用精确（prec）和召回（rec）计算：

*,* （3）

计算精度和召回率的方法是：

（4）

*,* （5）

哪里

•是真阳性的数量：地面真值标签显示这是异常，我们的算法将其正确分类为异常。*TP*

•是误报的数量：地面真值标签说这不是异常，但我们的算法错误地将其归类为异常。*计划生育*

•是假阴性的数量：地面真值标签说这是异常，但我们的算法错误地将其归类为非异常。*FN*

在提供的代码selectThreshold.m中，已经有一个循环尝试了许多不同的ε值，并根据f1分数选择最佳的ε。

现在您应该在selectThreshold.m中完成代码。您可以在所有交叉验证示例中使用for循环实现f1分数的计算（以计算tp、fp、fn值）。你应该看到epsilon的值是8.99e-05。

|  |
| --- |
| **实施说明：**为了计算tp、fp和fn，您可以使用矢量化实现，而不是在所有示例上循环。这可以通过octave/matlab的向量与单个数的相等性测试来实现。如果在n维二元向量v∈{0,1}n中有多个二元值，则可以使用：sum（=0）找出该向量中有多少个值是0。您还可以对此类二进制向量应用逻辑和运算符。例如，让cvpredictions是一个二进制向量，其大小与交叉验证集的数目相同，其中，如果算法考虑异常，则第i个元素为1，否则为0。例如，您可以使用：fp=sum（（cvpredictions==1）和（yval==0）计算误报数。*V* |

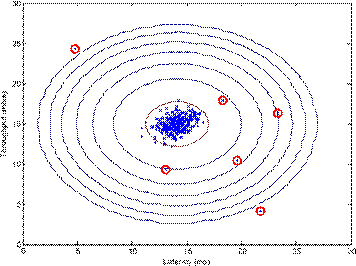


图3：分类异常。

完成selectThreshold.m中的代码后，ex8.m中的下一步将运行异常检测代码并在图中圈出异常（图3）。

*你现在应该提交你的解决方案。*

## 1.4           高维数据集

脚本ex8.m的最后一部分将运行您在更真实、更难处理的数据集上实现的异常检测算法。在这个数据集中，每个示例由11个特性描述，捕获计算服务器的更多属性。

脚本将使用您的代码来估计高斯参数（μi和σi2），计算估计高斯参数所依据的两个训练数据x的概率，并对交叉验证集xval执行此操作。最后，使用selectthreshold找到最佳阈值ε。

你应该看到一个值epsilon约为1.38e-18，并发现117个异常。



# 2             推荐系统

在练习的这一部分中，您将实现协作过滤学习算法，并将其应用于电影分级数据集。该数据集由1到5的分级组成。这个数据集有nu=943个用户，nm=1682个电影。在练习的这一部分中，您将使用脚本ex8[[ 2 ]](" \l "_ftn2" \o ") 科菲

在本练习的下一部分中，您将实现函数coficostfunc.m，该函数计算协作fitlering目标函数和渐变。在实现了成本函数和梯度之后，您将使用fmincg.m学习协同过滤的参数。

## 2.1           电影分级数据集

脚本ex8 cofi.m的第一部分将加载数据集ex8 movies.mat，在Octave/Matlab环境中提供变量y和r。

矩阵y（a nummovies×num users matrix）存储收视率y（i，j）

（从1到5）。矩阵r是二值指示矩阵，其中，如果用户j对电影i进行了评级，则r（i，j）=1，否则r（i，j）=0。协同过滤的目的是预测用户尚未评分的电影的评分，即r（i，j）=0的条目。这将允许我们向用户推荐预测收视率最高的电影。

为了帮助您理解矩阵y，脚本ex8cofi.m将计算第一部电影（玩具总动员）的平均电影评分，并将平均评分输出到屏幕。

在练习的这一部分中，您还将使用矩阵x和θ：

|  |  |
| --- | --- |
| （1））—*T*  -（x）  —（x（2））—x=…γ*T*γγ*,*  γ  -（x（n））。-*米T* | （1））—*T*  -（Th）  —（θ（2））—θ=…γ*T*γγ*.*  γ  -（θ（n））.-*UT* |

x的第i行对应于第i电影的特征向量x（i），θ的第j行对应于第j用户的一个参数向量θ（j）。x（i）和θ（j）都是n维向量。在本练习中，将使用n=100，因此，x（i）∈r andθ（j）∈r。相应地，x是nm×100矩阵，θ是nu×100矩阵。100 100

## 2.2           协同过滤学习算法

现在，您将开始实现协作过滤学习算法。您将从实现成本函数开始（不带正则化）。

电影推荐设置中的协同过滤算法考虑了一组n维参数向量x（1），…，x（nandθ（1），…，θ（n，其中模型预测用户j对电影i的评分为y（i，j）=（θ（j））tx（i）。给定一个数据集，该数据集由一些用户在某些电影上生成的一组分级组成，您希望了解参数向量*米*）*U*）

*X*（1）*，…，X*（n）*米*）*，Th*（1）*，…，Th*（n）*U*）产生最佳拟合（最小化平方误差）。

您将在coficostfunc.m中完成代码，以计算协同过滤的成本函数和梯度。请注意，函数的参数（即，您要学习的值）是x和θ。为了使用现成的最小化方法，如fmincg，我们设置了成本函数，将参数展开为单个向量参数。您以前在神经网络编程练习中使用过相同的矢量展开方法。

### 2.2.1          协同过滤成本函数

协作过滤成本函数（无正则化）由

*.*

现在，您应该修改coficostfunc.m以在变量j中返回此成本。请注意，仅当r（i，j）=1时，才应该累积用户j和电影i的成本。

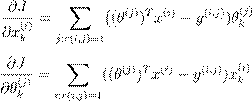
完成此函数后，脚本ex8 cofi.m将运行成本函数。你应该会看到22.22的输出。

*你现在应该提交你的解决方案。*

**实施说明：**我们强烈建议您使用矢量化实现来计算j，因为稍后优化包fmincg会多次调用它。像往常一样，首先编写一个非矢量化的实现（以确保您有正确的答案）可能是最简单的，然后修改它使其成为矢量化的实现（检查矢量化步骤不会更改算法的输出）。要想出一个矢量化的实现，下面的提示可能会有帮助：您可以使用r矩阵将所选条目设置为0。例如，r.\*m将在m和r之间执行元素相乘；因为r只有值为0或1的元素，所以只有当r中的相应值为0时，这才有将m的元素设置为0的效果。因此，sum（sum（r.\*m））是m的所有元素的和，r中的相应元素等于1。

### 2.2.2          协同过滤梯度

现在，您应该实现渐变（不带正则化）。具体来说，您应该完成coficostfunc.m中的代码以返回变量x毕业生和西塔毕业生。请注意梯度应该是与x大小相同的矩阵，同样地，θ梯度是与θ大小相同的矩阵。成本函数的梯度如下：

*.*

请注意，该函数通过将两组变量展开为一个向量来返回它们的梯度。完成计算渐变的代码后，脚本ex8cofi.m将运行一个渐变检查（checkCostFunction）来数值检查渐变的实现。如果实现正确，您应该发现分析渐变和数值渐变非常匹配。[〔3〕](" \l "_ftn3" \o ")

*你现在应该提交你的解决方案。*

|  |
| --- |
| **实施说明：**您可以在不使用矢量化实现的情况下获得此分配的全部学分，但您的代码将运行得慢得多（几个小时），因此我们建议您尝试将实现矢量化。  首先，可以使用for loop over movies（用于计算）和for loop over users（用于计算）实现渐变。当您第一次实现渐变时，您可以从一个未划分的版本开始，通过实现另一个内部for循环来计算求和中的每个元素。以这种方式完成渐变计算后，您应该尝试将实现矢量化（将内部for循环矢量化），这样您只剩下两个for循环（一个用于循环电影以计算每个电影，另一个用于循环用户以计算每个用户）。 |

|  |
| --- |
| **实施技巧：**要执行矢量化，您可能会发现这很有帮助：您应该想出一种方法来同时计算与之相关的所有导数（即，与特征向量x（i）相关的导数项）。我们将第i部电影的特征向量的导数定义为：  （XGRAD）  要将上述表达式矢量化，可以从索引到θ和y开始，只选择感兴趣的元素（即r（i，j）=1的元素）。直观地说，当你考虑第i部电影的特性时，你只需要关心给电影打分的用户，这允许你从θ和y中删除所有其他用户。  具体来说，您可以将idx=find（r（i，：）==1）设置为已对电影i进行分级的所有用户的列表。这将允许您创建索引到θ和y中的临时矩阵thetatep=theta（idx:）和ytemp=y（iidx），以便仅为已对第i部电影进行分级的用户集。这将允许您将衍生工具写成：*,,*  xgrad（i:）=（x（i:）\*thetatemp-ytemp）\*thetatemp*,,.*  （注意：上面的向量化计算返回的是行向量。）  在对x（i）的导数计算进行矢量化之后，还应该使用类似的方法对θ（j）的导数进行矢量化。 |

### 2.2.3          正则化成本函数

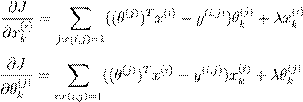
给出了正则化协同滤波的代价函数。

现在，您应该将正则化添加到成本函数j的原始计算中。完成后，脚本ex8 cofi.m将运行正则化成本函数，您应该会看到大约31.34的成本。

*你现在应该提交你的解决方案。*

### 2.2.4          正则梯度

既然已经实现了正则化成本函数，就应该继续实现梯度的正则化。您应该添加到coficostfunc.m中的实现中，通过添加来自正则化项的贡献来返回正则化梯度。注意，正则化成本函数的梯度由以下公式给出：

*.*

这意味着您只需将λx（i）添加到前面描述的x梯度（i，：）变量，并将λθ（j）添加到前面描述的θ梯度（j，：）变量。

完成计算渐变的代码后，脚本ex8 cofi.m将运行另一个渐变检查（checkCostFunction），以数字方式检查渐变的实现。

*你现在应该提交你的解决方案。*

## 2.3           学习电影推荐

在你完成了协同过滤成本函数和梯度的实现之后，你现在可以开始训练你的算法，为你自己制作电影推荐。在ex8 cofi.m脚本的下一部分中，您可以输入自己的电影首选项，以便以后运行算法时，您可以获得自己的电影推荐！我们已经根据自己的喜好填写了一些价值观，但是你应该根据自己的喜好来改变。可以在文件movie idx.txt中找到数据集中所有电影及其编号的列表。

### 2.3.1          建议

给你的最佳推荐：

电影《泰坦尼克号》预测评分9.0（1997）

电影《星球大战》（1977）预测评分8.9

电影《肖申克的救赎》预测评分8.8（1994）

预测电影的收视率为8.5（1997）

电影《寻欢作乐》预测评分8.5（1997）

电影常见嫌疑犯预测评分8.5（1995）

电影辛德勒排行榜预测评分8.5（1993）

《迷失方舟》电影《掠夺者》预测评分8.4（1981）

《电影帝国反击战》（1980）预测收视率为8.4

电影《勇敢的心》（1995）预测评分8.4

提供的原始评级：

《玩具总动员》（1995）4级

对12只猴子评定为3级（1995年）

对普通嫌疑犯评定为5级，1995年

爆发率为4级（1995年）

《肖申克救赎》（1994）5级

在你睡觉的时候被评为3级（1995）

《阿甘正传》5级（1994）

《沉默的羔羊》（1991）被评为2级

外星4级（1979）

硬质合金2（1990）评定为5级

Sphere等级5（1998）

图4：电影推荐

在将附加评级添加到数据集之后，脚本将继续训练协作过滤模型。这将学习参数x和θ。要预测用户j的电影i的等级，需要计算（θ（j））t x（i）。脚本的下一部分计算所有电影和用户的分级，并根据脚本前面输入的分级显示它推荐的电影（图4）。注意，由于不同的随机初始化，您可能获得不同的预测集。

# 提交和评分

完成任务的各个部分后，请务必使用提交功能系统将您的解决方案提交到我们的服务器。下面是这个练习的每个部分的得分情况。

|  |  |  |
| --- | --- | --- |
| **部分** | **提交的文件** | **点** |
| 估计高斯参数 | 估计Guassian.m | 15点 |
| 选择阈值 | 选择阈值.m | 15点 |
| 协同过滤成本 | coficostfunc.m公司 | 20点 |
| 协同过滤梯度 | coficostfunc.m公司 | 30点 |
| 正规化成本 | coficostfunc.m公司 | 10点 |
| 正则化梯度 | coficostfunc.m公司 | 10点 |
| 总分 |  | 100点 |

您可以多次提交解决方案，我们只考虑最高分数。

[〔1〕](" \l "_ftnref1" \o ")Octave是Matlab的免费替代品。对于编程练习，您可以自由使用倍频程或Matlab。

[[ 2 ]](" \l "_ftnref2" \o ")来自grouplens research的movielens 100k数据集。

[〔3〕](" \l "_ftnref3" \o ")这类似于你在神经网络练习中使用的数值检查。